LaTeX Workshop

Using LaTeX to produce your thesis

Dr Greg Sheard
July 2007

Today’s workshop

- Introduction to LaTeX
  - Producing documents
  - What is LaTeX?
  - LaTeX or Word?
  - Getting Started
  - LaTeX commands & examples
- Thesis Template
  - What is included?
  - Walkthrough

Producing documents

- What are documents?
  - Paper, form, letter, memo, book, etc.
    - i.e. Written material
  - Designed to convey information
- Documents can be short (shopping list), or long (20+ volumes of Oxford English Dictionary)

Producing documents

- How are documents generated?
  - Very short documents (≤ 1 page):
    - By hand
    - Dictated to secretary
    - Typed up using word processor
  - Shortish documents (1 < page < 20-30?):
    - Typically word processor used

Producing documents

- How are documents generated?
  - Longer documents (>20-30 pages):
    - Some form of typesetting/markup language employed
    - Word processors used (employing cross-referencing, style formatting, etc)
- Longer documents require more work if style, content, numbering or formatting changes required
  - Typesetting languages designed to do this

What is LaTeX?

- Computer program for typesetting documents
- Based on TeX (created by D.E. Knuth)
- Well-suited for long documents
  - Auto-numbering facilities for chapters, sections, theorems, equations.
  - Cross-referencing.

What is $\LaTeX$? (cont.)

- Different versions available:
  - $\LaTeX$
  - $\LaTeX$ 2
  - $\LaTeX$ 2ε
- Each extends functionality of previous version
- We will use $\LaTeX$ 2ε

$\LaTeX$ or Word?

- Word processing packages can do what $\LaTeX$ can do
  - Generate Table of Contents
  - Number sections, subsections, figures, tables, etc
  - Can use add-ons (e.g. End Note) to generate bibliography

$\LaTeX$ or Word?

- Word processing packages also
  - Show (almost) exactly what your document will look like on-screen
    - Known as WYSIWYG → "What You See Is What You Get"
  - Allow resizing and positioning of figures
  - Allow integration of spreadsheet-generated plots, etc.

$\LaTeX$ or Word?

- What is the problem with WYSIWYG word processors?
  - Inordinate file sizes
  - Loss of embedded figure quality
  - Bizarre & unstable section numbering & style handling
  - Appalling mathematics!
  - Limited cross-platform compatibility

$\LaTeX$ or Word?

- What is the problem with WYSIWYG word processors?
  - Drain on system resources
    - Large documents + lots of figures = long load/alteration/save times → Author frustration!
- Let's consider the advantages/disadvantages of our alternative: $\LaTeX$

$\LaTeX$ or Word?

- Advantages of $\LaTeX$:
  - Free
  - Available on many platforms (e.g. Windows, Linux, Mac)
    - Transport your thesis from your home Windows PC to your Linux machine at Uni
  - $\LaTeX$ works with ASCII files
    - Hence portable
    - Also, can see how others achieve particular results → mimic good ideas!

LaTeX or Word?

- Advantages of LaTeX:
  - Can use any text editor (e.g. VI, EMACS, MS Word ☺)
  - Superior typesetting (especially mathematics)
  - Style changes are straightforward
    - Journals supply their own style files, dictating how *everything* is handled and displayed

LaTeX or Word?

- Advantages of LaTeX:
  - Extensibility – most of what you might want to do has been done – the solutions/code fragments/packages are online & free
    - Examples:
      - Spread tables over several pages
      - Include figure formats other than standard .eps
      - Extend referencing capabilities
      - Force section figures to appear after section headings

LaTeX or Word?

- Disadvantages of LaTeX:
  - Font selection difficult - LaTeX built around a limited number of standard fonts:
    - A roman font
    - `A typewriter font`
    - A sans serif font
    - **Bold**, *italic*, *slanted* and SMALL CAPS varieties
    - Selecting, say, this Arial font can require additional packages!

LaTeX or Word?

- Disadvantages of LaTeX:
  - Has trouble flowing text around figures
    - Much easier just to keep text above & below
  - Separation of style from content
    - You might set your heading font & spacing in one file,

      yet write the text in a different file
    - This takes getting used to for non-programmer types…

LaTeX or Word?

- Disadvantages of LaTeX:
  - No graphical front end
    - To see what your document (or one tiny alteration) looks like, must
      - Recompile
      - View generated document using separate application
      - Can be time consuming if your system & applications are not set up appropriately
        - There are some text editors designed for use with LaTeX → can compile & display with one click

Getting Started

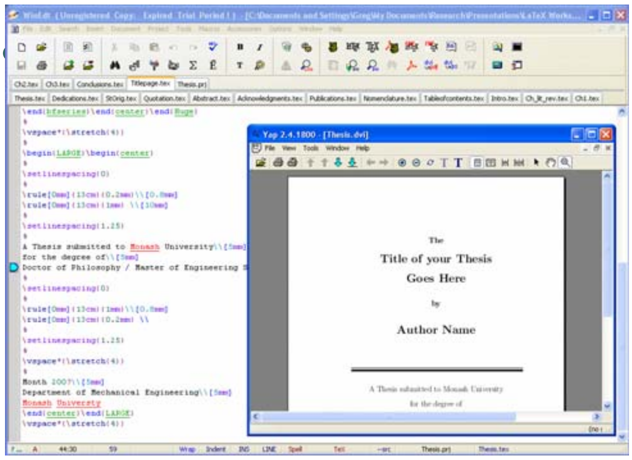
- What you will need:
  - A LaTeX 2ε package
    - Linux versions online
    - Windows: MikTeX is a good option
  - A text editor
    - Linux: VI, Joe, EMACS, etc…
    - Windows:
      - Shareware version of WinEdt, a text editor tailored for LaTeX usage
        - Can edit, compile & convert documents

www.miktex.org

MiKTEX

WinEdt Shell

www.winedt.com



```
\end{bfseries}\end{center}\end{Huge}
%
\vspace*{\stretch{4}}
%
\begin{LARGE}\begin{center}
%
\setlinespacing{0}
%
\rule[0mm]{13cm}{0.2mm}\\[0.8mm]
\rule[0mm]{13cm}{1mm} \\[10mm]
%
\setlinespacing{1.25}
%
A Thesis submitted to Monash University\\[5mm]
for the degree of\\[5mm]
Doctor of Philosophy / Master of Engineering
%
\setlinespacing{0}
%
\rule[0mm]{13cm}{1mm}\\[0.8mm]
\rule[0mm]{13cm}{0.2mm} \\
%
\setlinespacing{1.25}
%
\vspace*{\stretch{4}}
%
Month 2007\\[5mm]
Department of Mechanical Engineering\\[5mm]
Monash Univercty
\end{center}\end{LARGE}
\vspace*{\stretch{4}}
```

The

Title of your Thesis

Goes Here

by

Author Name

A Thesis submitted to Monash University

for the degree of

Yap 2.4.1800 - [Thesis.dvi]
File View Tools Window Help
Ch2.tex Ch3.tex Conclusions.tex Titlepage.tex Thesis.prj
Thesis.tex Dedications.tex StOrig.tex Quotation.tex Abstract.tex Acknowledgments.tex Publications.tex Nomenclature.tex Tableofcontents.tex Intro.tex Ch_lit_rev.tex Ch1.tex
44:30 59 Wrap Indent INS LINE Spell

Getting Started

- What you will need:
  - Install packages for Windows versions of the required software will be made available with this workshop
  - More recent versions may be available online
    - Note: MikT$_E$X desires Internet access to self-update or to auto-download L^{A}T$_E$X packages not available locally

A Simple Example

- See `example1.tex`
- Note the statements bracketing the document:

```
\documentclass[a4paper,12pt]{article}
\begin{document}
```

and

```
\end{document}
```

- These define the type of document, as well as its beginning and end

Special Characters

- Most of the other characters are just text
- Exceptions are the *special characters* \, $, { and } (other *special characters* include &, ^, _, %, ~, #.
  - Typing these will not produce the character
  - To generate the character in text, add \
    - E.g. For &, use \&

Special Characters

- Exceptions are \, ^, ~
  - use `\char92` for "\",
  - `\char94` for "^",
  - and `\char126` for "~" (the numbers are the ASCII codes for these characters)

Special Characters

- The characters { and } are *grouping* characters
  - Anything within them is treated as one unit
  - E.g. In the example we use these to delineate changes of font

Special Characters

- The character $ is used to enclose mathematical expressions in text
  - i.e. “`$D$ and D`” produces the output “*D* and D”
  - Note: “`\(`” and “`\)`” produce the same effect

Control Sequences

- The example illustrated LaTeX control sequences – words directly following a “\” – these instruct LaTeX to produce an effect
  - E.g. `\textit{}` outputs the contents of the { } in *italic* font
  - E.g. `\textbf{}` outputs the contents of the { } in **boldface** font

Control Sequences

- Control sequences can also generate characters
  - E.g. `\epsilon` outputs ε,
  - `\delta` outputs δ,
  - `\in` outputs ∈
  - `\colon` outputs :
  - `\to` outputs →

Equations

- Equations requiring their own line can be generated by enclosing the statement between “`\[`” and “`\]`”
  - Note: The $ are basic TeX
- LaTeX also introduces the *equation environment* seen in the example
  - This permits equation numbering, labeling and referencing

Environments

- In LaTeX, environments are defined by specifying

```
\begin{environment_name}
    … … …
\end{environment_name}
```

  - Hence the *document* is a type of environment
  - Others include *figure*, *table*, *equation*, as well as user-defined environments

Equation environment

- Several environments exist in LaTeX for displaying equations, with *equation* being the simplest
  - The others allow multi-line, array and aligned equations
- Our example included an *equation* environment
  - Note the equation number produced

Naming an environment

- Naming our environment
  - Use `\label{…}` control sequence
  - Include “`\label{`*our_eqn_name*`}`” within environment
  - E.g. See `example2.tex`
  - We now have

```
\begin{equation}
\label{our_equation}
 |f(y) = f(x)| < \epsilon.
\end{equation}
```

Referencing an environment

- Referring to our environment
  - We use the `\ref{…}` control sequence
  - In `example2.tex`, the text “`With equation~\ref{ our_equation }, one may…`”
  - Produces the output

    With equation 1, one may…
  - This can be used for *figure*, *table* environments also (with each numbered separately

Displaying simple text

- See `example3.tex`
- Paragraphs are denoted by a blank line in your input `.tex` file
- Paragraph indentation is automatically handled
- Unlike Word, LaTeX requires left & right quotes to be specified using the “`` ` ``” and “`'`” characters, respectively
  - Use “`` ` ``” for single → `Blah…
  - Use “`` ` ` ``” for double → “Blah…

Displaying simple text

- Hyphens are generated using “`-`”
- Longer dashes are generated by using “`--`” and “`---`”
- LaTeX regards tabs and carriage returns (enters) as blank spaces
  - You need to explicitly enforce a new line
- LaTeX automatically leaves a little more space after “`,`”, “`:`”, “`;`” and “`.`” characters
- Spaces after control sequences are ignored

Displaying simple text

- For doubling up of quotes

  E.g. “LaTeX is ‘fun’”

  Use “`\,`” between each quote designation

Sections and referencing

- See `example4.tex`
- The example illustrates several depths of sections, displayed in the default fashion
  - Advanced users can alter how the headings are displayed

Font appearance & accents

- See `example5.tex`
- Example describes
  - Font size selection
  - Font shapes
  - Font families
  - Accents
- Note: Accents cannot be employed using these control sequences in equations

Superscripts & subscripts

- Invoked with the control characters ^ and _
- See `example6.tex`

Greek letters

- Control sequences named for each Greek letter
  - E.g. `\mu` produces *μ*
- Capital letters are achieved by capitalising the first letter in the control sequence
  - E.g. `\Sigma` produces *Σ*
- See `example7.tex`

Tables

- The example introduced the `table` environment
  - It “floats” a `tabular` environment structure
  - It contains a caption which are numbered
    - Labels must be placed in captions for referencing
- The `tabular` environment creates the table structure

Tables

- The `tabular` environment inputs are
  - `{tabular}` → the type of environment
  - {*column-formatting*} → Specifies the number of columns, and their alignment
  - E.g. `{lcclrr}` produces a 6-column table, with left / centred / centred / left / right / right alignment
  - Enter content row by row
    - Use the column separator character “&” between columns
    - Use the row separator “\\” at the end of each row

Mathematical characters

- Many characters are available, including:
  - Misc symbols (e.g. √, ∞, ∆)
  - Large operators (e.g. ∩, ∫, ∑)
  - Binary operators (e.g. *, ±, •)
  - Relations/negated relations (e.g. ∥, ≈, ≡, ≤)
  - Arrows, braces, etc.
- These are extensively tabulated online

Mathematical font selection

- Cannot use control sequences like `\textit`, `\textbf`...
- Instead (LaTeX 2ε only), use `\mathit`, `\mathbf`
- Also a “calligraphic” font invoked by `\cal`

Mathematical functions

- Fns such as sin, cos, exp, ln, lim, etc:
  - Many have named control sequences
  - Again these are well documented
    - E.g. “cos” is achieved by using `\cos`
  - If predefined ctrl sqnce not available, use `\mathrm{`*func_name*`}`
    - E.g. “cosec(x)” obtained by typing `\mathrm{cosec}(x)` in a mathematical expression

Text in mathematical expressions

- To embed text in a mathematical expression, use `\mbox{`*embedded_text*`}`
  - E.g. To get “$x + y = 1 \text{ for all } x > 3$” type `$x+y=1\mbox{ for all }x>3$`
  - Note the blank spaces before and after “for all” in the mbox
  - Without this, LaTeX would output “$x + y = 1 \text{for all} x > 3$”, as spacing is determined by mathematical typesetting rules

Fractions and roots

- Functions exist to typeset fractions and roots
- These can be compounded at will
  - Fractions: use `\frac{`*numerator*`}{`*denominator*`}`
  - Roots: use `\sqrt{`*argument*`}`
- E.g. To get $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

  use `$x=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$`

Brackets in mathematics

- If we just use the regular `\{`, `[`, `(` style brackets, the sizing might not be correct
  - E.g. `$x=(\frac{-b\pm\sqrt{b^2-4ac}}{2a})$` produces

$$x = (\frac{-b \pm \sqrt{b^2 - 4ac}}{2a})$$

Brackets in mathematics

- Instead we need to use
  - `\left(` and `\right)`
  - or for braces `\left{` and `\right}`
  - or square brackets `\left[` and `\right]`
  - or no brackets `\left.` and `\right.`
    - Can mix and match, but need to balance number of left & right
- E.g. Use `$x=\left(\frac{-b\pm\sqrt{b^2-4ac}}{2a}\right)$` to get

$$x = \left(\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}\right)$$

Multiline formulae

- Say you want

$$\begin{aligned} x &= \cos^2\theta + \sin^2\theta \\ &= 1 \end{aligned}$$

- This can be generated using the `eqnarray*` environment (`*` suppresses eqn. numbering)
  - Use alignment characters as w/`tabular` env.
  - E.g. For the above, use

```
\begin{eqnarray*}
X & = & \cos^2\theta + \sin^2\theta \\
& = & 1 \\
\end{eqnarray*}
```

Arrays & matrices

- Arrays or matrices can be displayed using `array` environment
  - Uses same separators as `tabular` and `eqnarray*` environments
  - E.g. See `example8.tex`

Derivatives

- Best to use `\frac` and roman fond for “d”s:
- i.e. `\frac{\mathrm{d}y}{\mathrm{d}x} = \frac{\partial y}{\partial x}` gives

$$\frac{\mathrm{d}y}{\mathrm{d}x} = \frac{\partial y}{\partial x}$$

- Convenient to set up user-defined fuctions to generate these (see thesis template)

Limits, sums, integrals

- Expressions such as lim, inf, sup, and ∑ often need content placed directly below or above the symbol/word
  - This is achieved using `^` and `_`
- Integral expressions are set up similarly
- E.g. See `example9.tex`

Whitespace

- Horizontal space: use `\hspace{`*dist*`}`
  - *dist* can be expressed in many units
  - LaTeX recognises pt (point), pc (pica), in (inch), bp (big point), cm, mm, dd (didot point), cc (cicero) & sp (scaled point)
  - E.g. for a 20mm horizontal gap, use `\hspace{20mm}`

Whitespace

- Vertical space: use `\vspace{`*dist*`}`
  - Used between paragraphs
  - If page breaks then no space added (use `\vspace*{`*dist*`}` instead)
- Full stops:
  - LaTeX automatically adds extra space after full stops as this makes it easier for reader to distinguish sentences
  - Use `\` after a full stop if you want to keep smaller spacing (i.e. for “*J. Fluid Mech.*” you would enter `\textit{J.\ Fluid Mech.}`

User-defined control sequences

- Defined using `\newcommand{`*name*`}{`*instructions*`}`
- E.g. For a command `\dydx`, to generate

$$\frac{\mathrm{d}y}{\mathrm{d}x}$$

we can specify before the `\begin{document}` command:

```
\newcommand{\dydx}{\frac{\mathrm{d}y}{\mathrm{d}x}}
```

User-defined control sequences

- We can also pass n expressions to the command, e.g. , `\newcommand{`*name*`}[`*n*`]{`*instructions*`}`
- E.g. For a command `\dde`, which displays a derivative of *expr1* w.r.t. *expr2*:

$$\frac{\mathrm{d}(expr1)}{\mathrm{d}(expr2)}$$

we can specify:

```
\newcommand{\dde}[2]{\frac{\mathrm{d}#1}{\mathrm{d}#2}}
```

Thesis template

- What is included?
  - Main input file: `Thesis.tex`
  - WinEdt project file: `Thesis.prj`
  - Various style files (`.sty`)
  - Bibliography file (`Thesis.bib`)
    - Bibliography style file (`jfm_mod.bst`)
  - Chapter input files (`.tex`):
    - In `Chapter` & `Preface` subdirectories
  - Figure files (`.eps`)
    - In `Figs` subdirectory

Thesis main input file

- Contents:
  - Preamble
    - Packages used
    - Bibliography, page layout
    - User-defined commands, environments, definitions, control sequences
  - Document
    - Preface (note spacing & numbering format)
    - Intro (Roman numbering eqn, figs, tables)
    - Body (Arabic numbering)
  - Bibliography

Thesis template subfolders

- Preface:
  - .tex files for early pages in thesis
- Chapters:
  - .tex files for each chapter, including *Introduction*, *Literature Review* & *Conclusions*
- Figs:
  - .eps figure files for each chapter
    - Good idea to have separate folder for each – keeps this better-organised

Good reference books

- LaTeX will be easier to use with reference books such as

  - Goossens, M., Mittlebach, F. & Samarin, A. (1994) The LaTeX Companion, *Addison-Wesley*.
  - Kopka, H. & Daly, P.W. (1999) A Guide to LaTeX, *Addison-Wesley*.